Uniform Distribution

Definitions:

- Uniform Distribution: The probability of each valid outcome is the same.
- Geometric Distribution: The number of Bernoulli trials until you get a success
- Pascal Distribution: The number of Bernoulli trials until you get r successes

distribution	description	pdf	mgf	mean	variance
Uniform range = (a,b)	toss an n-sided die	$\left(\frac{1}{1+b-a}\right)\sum_{k=a}^{b}\delta(k-m)$	$\left(\frac{1}{n}\right)\left(\frac{1+z+z^2+...+z^{n-1}}{z^{n-1}}\right)$	$\left(\frac{a+b}{2}\right)$	$\left(\frac{(b-a+1)^2-1}{12}\right)$
Geometric	Bernoulli until 1st success	$(1-p)^{k-1}p$	$\left(\frac{(1-p)z}{z-p}\right)$	$\left(\frac{1}{p}\right)$	$\left(\frac{1-p}{p^2}\right)$
Pascal	Bernoulli until rth success	$\binom{k+r-1}{k}(1-p)^r p^k$	$\left(\frac{(1-p)z}{z-p}\right)^r$	$\left(\frac{r}{p}\right)$	$\left(\frac{r(1-p)}{p^2}\right)$

source: Wikipedia

Uniform Distribution:

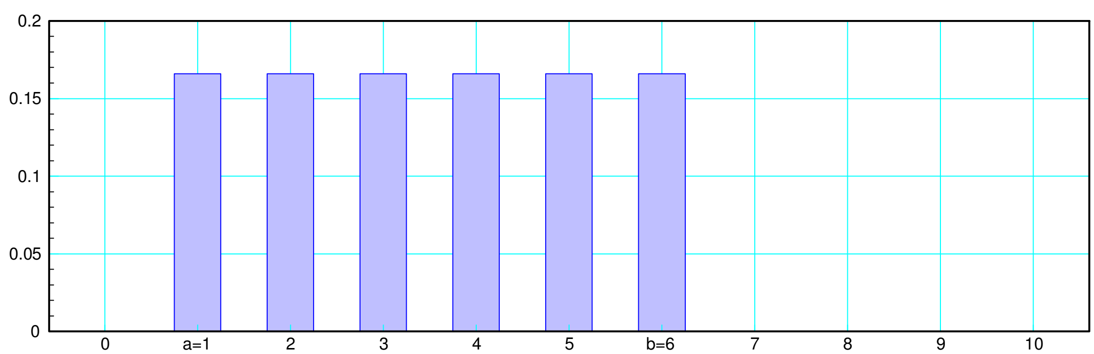
A uniform distribution can be thought of as an extension of a Bernoulli trial where

- There are n possible outcomes (rather than just two), and
- All outcomes have the same probability.

In general the pdf for a uniform distribution over the range of (a, b) (where n = 1+b-a = the number of possible outcomes) is

$$X(m) = \left(\frac{1}{n}\right)\sum_{k=a}^{b}\delta(k-m)$$

For example, for a 6-sided die (a=1, b=6), the pdf would be:



pdf for a 6-sided die

The z-transform for a uniform distribution over the range of [a, b] is

$$X(z) = \left(\frac{1}{6}\right)\left(\frac{1+z+z^2+z^3+z^4+...+z^{b-a}}{z^b}\right)$$

The mean is the average:

$$\mu = \left(\frac{a+b}{n}\right)$$

and the variance is approximately

$$\sigma^2 = \left(\frac{(b-a+1)^2-1}{12}\right)$$

There are many examples of uniform distributions, such as

- Drawing a card from a deck of cards (each has a probability of 1/52)
- A number coming up in Roulette (1 in 31 in Vegas, 1 in 32 in Atlantic City)
- A number coming up in the lottery (1 in 78,960,960)
- Being selected for jury duty (1 in 15,000. Ten people are selected from a county population of 150,000)

There are some betting schemes which take advantage of processes which are supposed to be uniform but are not. For example, about 15 years ago, someone watched which numbers came up in Roulette in Vegas and found that some numbers were more common that others. He/she (I forget which) won money with this scheme. Now, the roulette wheels are mixed every night (under tight security) and you are not allowed to watch and take notes.

Dungeons & Dragons

If you're a fan of *Dungeons and Dragons*, you're use to combining several types of die rolls. Different types of spells do different amounts of damage - with the total damage being the sum of various die rolls. In D&D, the notation for rolling dice is

XdY roll X dice, each die has Y sides.

For example, the notation 4d6 means *the sum of four 6-sided dice*. From *listfist.com*, some D&D spells are:

Spell Name	Level	Damage Type	Damage
Frostbite	0	Cold	1d6
Thunderous Smite	1	Thunder	2d6
Mind Whip	2	Psychic	3d6
Thunder Step	3	Thunder	3d10
Ice Storm	4	Bludgeoning + Cold	2d8 + 4d6
Chain Lightning	6	Lightning	10d8
Meteor Swarm	9	Bludgeoning + Fire	20d6 + 20d6

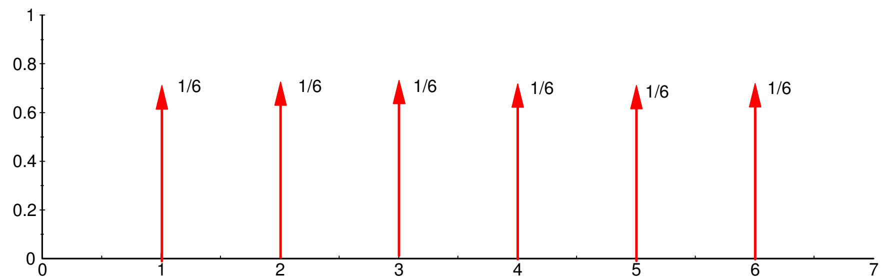
Frostbite (1d6)

- Druid, sorcerer, warlock, wizard
- Level 0
- 1d6 cold damage

For example, a fair six sided die would have six possible outcomes, each with probability of 1/6

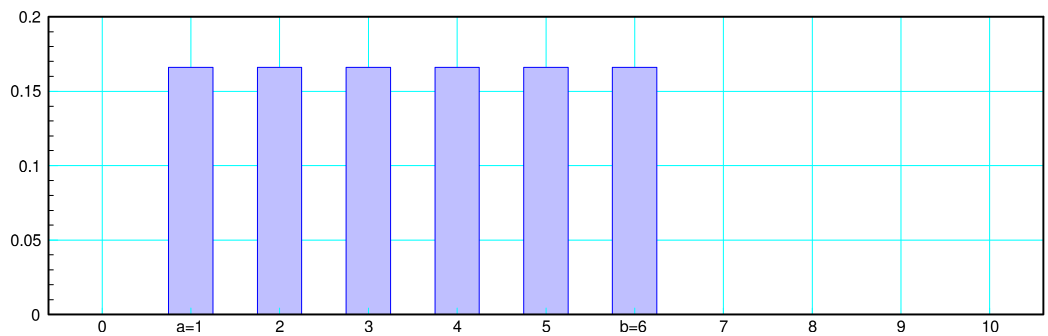
Die Roll: m	0	1	2	3	4	5	6	7+
Probability: X(m)	0	1/6	1/6	1/6	1/6	1/6	1/6	0

The pdf for this would be a delta function at each integer value:



pdf for a fair 6-sided die

Often times, this is represented using a bar graph with the understanding that the pdf is only non-zero at the integer values



pdf for a 6-sided die (again)

The mean and variance of a fair 6-sided die (d6) is

$$\mu = \left(\frac{1+2+3+4+5+6}{6} \right) = 3.5$$

$$\sigma^2 = \left(\frac{(1-3.5)^2+(2-3.5)^2+(3-3.5)^2+(4-3.5)^2+(5-3.5)^2+(6-3.5)}{6} \right)$$

$$\sigma^2 = 2.91667$$

Thunderous Smite (2d6)

- Paladin spell
- Level 1
- Weapon does an additional 2d6 of thunder damage

There are several ways to compute the probability of doing N damage when rolling two 6-sided dice.

Monte-Carlo: Write a Matlab program to

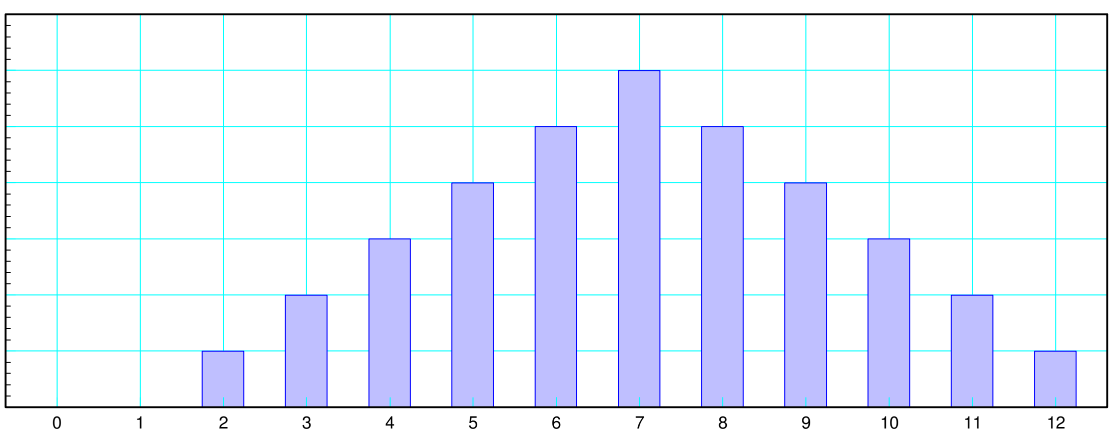
- Find the sum of two 6-sided dice (2d6)
- Record the outcome of each roll, and
- Repeat one million times.

Matlab Code:

```
Damage = zeros(12,1);
for k=1:1e6
    N = sum( ceil(6*rand(1,2)));
    Damage(N) = Damage(N) + 1;
end
k = [1:12]';
[k,Damage/1e6 * 36]
```

Result:

```
Damage   p*36
  1        0
  2     0.9905
  3     1.9993
  4     3.0073
  5     3.9918
  6     5.0194
  7     6.0026
  8     5.0032
  9     3.9948
 10     2.9965
 11     1.9959
 12     0.9986
```



pdf for Thunderous Smite (2d6)

Enumeration:

Go through every combination of two 6-sided dice and keep track of the results.

```
Damage = zeros(12,1);
for d1=1:6
    for d2 = 1:6
        Roll = [d1, d2];
        N = sum(Roll);
        Damage(N) = Damage(N) + 1;
    end
end
k = [1:12]';
[k,Damage]
```

The results match up with Monte Carlo results, only more exact

Damage	M-Carlo	Enumeration
1	0	0
2	0.9905	1
3	1.9993	2
4	3.0073	3
5	3.9918	4
6	5.0194	5
7	6.0026	6
8	5.0032	5
9	3.9948	4
10	2.9965	3
11	1.9959	2
12	0.9986	1

Moment Generating Functions (z-Transforms)

When you add random variables,

- You convolve the pdf's, or
- You multiply the moment generating functions

The z-transform for a single 6-sided die is

$$\psi(z) = \frac{1/6}{z} + \frac{1/6}{z^2} + \frac{1/6}{z^3} + \frac{1/6}{z^4} + \frac{1/6}{z^5} + \frac{1/6}{z^6}$$

This can be re-written as

$$\psi(z) = \left(\frac{1}{6z^6}\right)(1 + z + z^2 + z^3 + z^4 + z^5)$$

When you convolve pdf's, you multiply the z-transform. Hence, the z-transform for 2d6 is

$$\psi(z) = \left(\frac{1}{6z^6}\right)^2 (1 + z + z^2 + z^3 + z^4 + z^5)^2$$

The first term is easy. The second term is convolution. You really can't avoid convolution.

Convolution: Since you're stuck with convolution, you might as well live with it. In Matlab, the pdf for a 6-sided die (scaled by 6), is:

```
d6 = [0,1,1,1,1,1,1]

d6 =      0      1      1      1      1      1      1
```

Two six-sided dice will be (scaled by $6^2 = 36$)

```
d6x2 = conv(d6,d6)

d6x2 =   0    0    1    2    3    4    5    6    5    4    3    2    1
```

This matches the previous results as well - meaning you can use any of these approaches when summing dice:

- Monte-Carlo
- Enumeration
- Convolution
- z-Transform (which is the same as convolution)

They all give the same results (Monte-Carlo is less accurate, however).

Mean and Variance

The mean and variance can be found several ways. First, when you add two random variables

- The means add and
- The variance adds

These can be found by scaling the mean and variance for a single die:

$$\mu = 2 \cdot 3.5 = 7$$

$$\sigma^2 = 2 \cdot 2.9167 = 5.8333$$

These can also be found using a Monte Carlo simulation::

```
Data = [];
for n=1:1e3
    Roll = ceil(6*rand(1,2));
    N = sum(Roll);
    Data = [Data; N];
end
x = mean(Data)
v = var(Data)

x  =     6.9780
v  =     5.8253
```

Convolution also works:

```
d6 = [0,1,1,1,1,1,1]' / 6;
D  = conv(d6, d6);
k  = [0:length(D)-1]';
x = sum(D .* k)
v = sum(D .* (k - x).^2)

x =      7
v =    5.8333
```

This also tells you that when summing dice, the mean and variance can be found several ways:

- Scale the mean and variance of rolling a single die, or
- Compute the mean and variance from a Monte-Carlo simulation, or
- Compute the mean and variance from the pdf, found from convolution

They all give the same results.

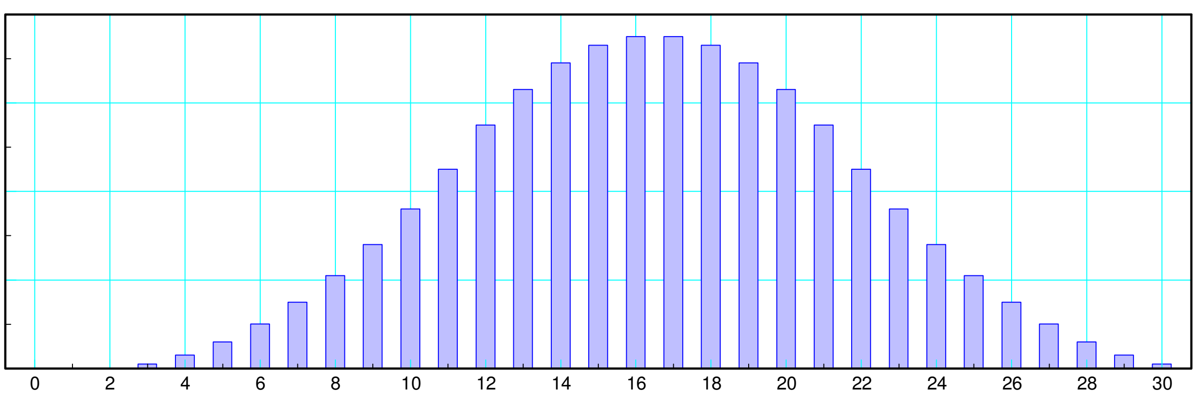
Thunder Step (3d10)

- Sorcerer Spell
- Level 3
- 3d10 of thunder damage

The damage for the sum of 3d10 can be found using convolution

```
d10 = [0,1,1,1,1,1,1,1,1,1,1]' /10;
d10x2 = conv(d10, d10);
d10x3 = conv(d10, d10x2);
D = d10x3;
k  = [0:length(D)-1]';
x = sum(D .* k)
v = sum(D .* (k - x).^2)

x =   16.5000
v =   24.7500
```



pdf for 3d10 (Thunderous Step)

The mean and variance can also be found by scaling. For a single die

$$\mu = \left(\frac{a+b}{2}\right) = \left(\frac{1+10}{2}\right) = 5.5$$

$$\sigma^2 = \left(\frac{(b-a+1)^2-1}{12}\right) = \left(\frac{10^2-1}{12}\right) = 8.25$$

The sum of 3d10 therefore has

$$\mu = 3 \cdot 5.5 = 16.5$$

$$\sigma^2 = 3 \cdot 8.25 = 24.75$$

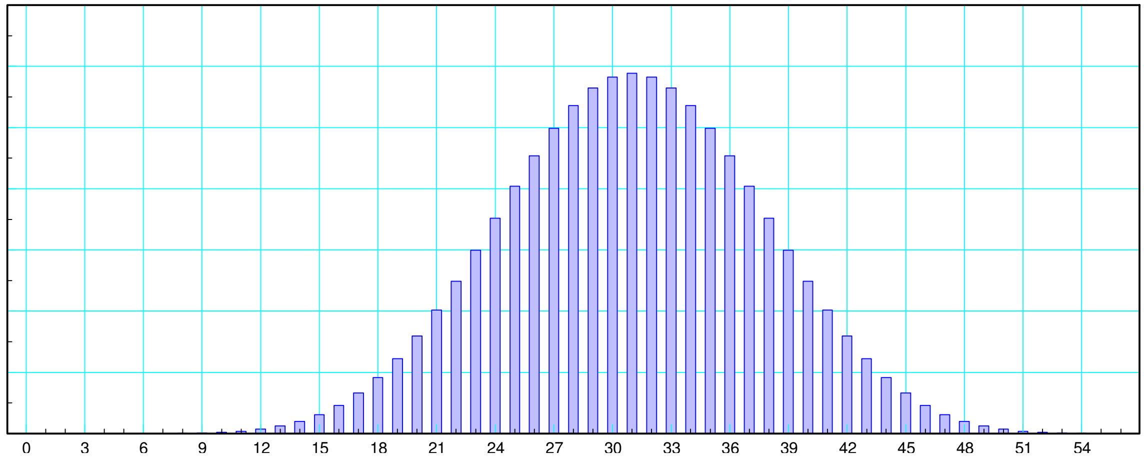
Ice Storm

- Sorcerer / Wizard / Druid Spell
- Level 4
- 2d8 bludgeoning damage plus 4d6 cold damage

The pdf for 2d8 + 4d6 can be found using convolution in Matlab:

```
d6 = [0,1,1,1,1,1,1]'/6;
d6x2 = conv(d6, d6);
d6x4 = conv(d6x2, d6x2);
d8 = [0,1,1,1,1,1,1,1,1]'/8;
d8x2 = conv(d8,d8);
D = conv(d6x4, d8x2);
k  = [0:length(D)-1]';
x = sum(D .* k)
v = sum(D .* (k - x).^2)

x =   23.0000
v =   22.1667
```



pdf for the damage done by an Ice Storm spell (4d6 + 2d8)

The mean and variance can be found by scaling these values from a single die:

d6:

$$\mu = \left(\frac{1+6}{2}\right) = 3.5$$

$$\sigma^2 = \left(\frac{(6-1+1)^2-1}{12}\right) = 2.91667$$

d8:

$$\mu = \left(\frac{1+8}{2}\right) = 4.5$$

$$\sigma^2 = \left(\frac{(8-1+1)^2-1}{12}\right) = 5.25$$

4d6 + 2d8

$$\mu = 4 \cdot 3.5 + 2 \cdot 4.5 = 23.00$$

$$\sigma^2 = 4 \cdot 2.91667 + 2 \cdot 5.25 = 22.16666$$

These numbers match results from convolution.

German Tank Problem

- https://en.wikipedia.org/wiki/German_tank_problem

Switching gears a bit - another statistical problem related to the uniform distribution is called *The German Tank Problem*.

German Panther tank: The production number is printed on the turret

This type of problem is as follows:

Assume

- A population of N items exists, with each item having a clearly marked number from 1..N.
- M items are sampled from this population at random, with each element having equal probability of being selected (i.e. a uniform distribution)

What is the value of N given these M samples?

History: The origin of this problem goes back to World War II. The Allies had a problem with German Tiger and Panther tanks: the tanks used by Britain and The United States could not stand up against these tanks. If, as allied intelligence reported, Germany was producing thousands of these tanks, the Allied invasion of France would have to wait until more heavy tanks could be built and shipped to England. If, on the other hand, Germany only had a few hundred of these tanks, the Sherman medium tank would be good enough to win the war.

Both the Tiger and Panther tanks had their production number painted clearly on their side. This allowed allies to note these numbers (collect samples from population of size N). In addition, captured tanks had numbered transmission boxes and numbered wheels (more samples from a finite population).

From this data, what is the estimate of the total number of German Tiger and Panther tanks?

Result: This was the problem presented to statisticians back in WWII. Two different solutions were

Frequentist Approach:

$$N \approx m + \frac{m}{k} - 1$$

Baysian Approach:

$$N \approx m + \frac{m \ln(2)}{k-1}$$

Baysian Confidence Interval

$$N \approx \mu \pm \sigma$$

where

$$\mu = (m-1)\left(\frac{k-1}{k-2}\right)$$

$$\sigma^2 = \frac{(k-1)(m-1)(m-k+1)}{(k-3)(k-2)^2}$$

where

N is the population size (highest number)

m is the largest number from the sampled data

k is the number of samples

Historical Results: The monthly production of German tanks were estimated using

- Statistics,
- Allied Intelligence, and
- Actual production numbers obtained after the war.

The resulting numbers were (https://en.wikipedia.org/wiki/German_tank_problem)

Month	Statistics	Intelligence	German Records
June 1940	169	1,000	122
June 1941	244	1,550	271
Aug 1942	327	1,550	342

It turns out, the estimated based upon captured and observed German tanks was almost dead on. This allowed the Allies to proceed with D-Day: there were not enough German tanks to change the outcome of the war.

Example: Generate 10 samples from a population with a random unknown size (N):

```
>> N = round(200*rand) + 50;
>> X = rand(1,N);
>> [a,b] = sort(X);
>> Sample = b(1:10)
Sample =   122    98   173    31    39    21   174    33   151    91
```

From this sample, estimate the value of N.

Solution: For this sample

```
>> m = max(Sample)
m =   174

>> k = length(Sample)
k =    10
```

The estimates of N are then

Frequentist Approach:

```
>> N1 = m + m/k - 1
N1 =  190.4000
```

Baysian Approach:

```
>> N2 = m + m*log(2)/(k-1)
N2 =  187.4008
```

Baysian Confidence Interval

```
>> N3 = (m-1)*(k-1)/(k-2)
N3 =  194.6250

>> s2 = (k-1)*(m-1)*(m-k-1) / ( (k-3)*(k-2)^2 )
s2 =  566.4978

>> s = sqrt(s2)
s =   23.8012
```

```
>> Upper = N3+s
Upper =  218.4262

>> Lower = N3-s
ans =  170.8238
```

The actual population size is:

```
>> N
N =   213
```

Summary

Uniform distributions have equal probability for all possible outcomes. A typical example is the result of rolling an N-sided die. When you add dice together, you are actually convolving the pdf's for each die rolled. Various combinations give the result of various spells in the game Dungeons and Dragons.

In addition, if you have samples of a numbered population, you can estimate the size of the population using solutions from *the German Tank Problem.*